

IN THE CLAIMS

1. (Currently Amended) A system for encapsulating a business process workflow, comprising:

a controller in communication with a user interface, the controller operable to:

receive a request from the user interface;

identify a process module associated with the request; and

route the request to the process module to invoke the process module; and

a process the process module having a plurality of states, each state comprising logic defining a portion of a business process and comprising an identifier of a corresponding view to be presented to a user, the view identifier comprising a logical specification of the corresponding view to be presented, the process module operable to:

identify the state of the plurality of states corresponding to the request; and

provide the view identifier corresponding to the identified state to the controller;

~~a controller in communication with the process module and a user interface, wherein the controller translates user input and invokes the process module in response thereto, and wherein the controller further operable to:~~

~~receives~~ receive the view identifiers identifier from the process module module; and

~~generates views~~ generate a view for the user that ~~are~~ is based on the logical ~~specifications~~ specification of the ~~views~~ view in the view identifiers identifier and ~~are~~ is compatible with the user interface.

2. (Previously Presented) The system of Claim 1, wherein the controller comprises:

a content engine connected to the interface for receiving user inputs and invoking the process module in response thereto, and for receiving the view identifier therefrom;

a channel adapter connected to the content engine for receiving the view identifier from the content engine, and selecting a presentation to be generated for the user, and connected to the user interface for communicating the presentation to the user.

3. (Currently Amended) The system of Claim 1, further comprising:

a second controller in communication with a second user interface and with the process module, wherein the second controller translates user input from the second user interface and invokes the process module in response thereto, and wherein the second controller further receives the view ~~identifiers~~ identifier from the process module and generates ~~views~~ a view for the user that ~~are~~ is based on the logical ~~specifications~~ specification of the ~~views~~ view in the view ~~identifiers~~ identifier and ~~are~~ is compatible with the second user interface.

4. (Currently Amended) A method for responding to a user request received over a channel, comprising the steps of:

providing a controller in communication with a user interface;

providing a process module having a plurality of states, each state comprising logic defining a portion of a business process and comprising an identifier of a corresponding view to be presented to a user, the view identifier comprising a logical specification of the corresponding view to be presented;

~~receiving the user input over the channel;~~

~~sending the user input to the process module;~~

receiving at the controller a request from the user interface;

identifying at the controller the process module, the process module associated with the request;

routing the request from the controller to the process module to invoke the process module;

within the process module, changing a state thereof in response to the request and generating an identifier of a view corresponding to the state to be presented to the user;

receiving at the controller the view identifier from the process module;

selecting a view to be presented to the user that is based on the logical specification of the view in the view identifier and compatible with the channel; and

sending the view to the user over the channel.

5. (Previously Presented) The method of Claim 4, further comprising when changing state within the process module, accessing a business application software module to determine which view identifier to generate.

6. (Previously Presented) The method of Claim 4, further comprising when changing state within the process module, accessing a database.

7. (Previously Presented) The method of Claim 4, further comprising when changing state within the process module, modifying data in a database.

8. (Currently Amended) A method for communicating with a user of a computer system, comprising:

~~receiving input from a user interface;~~

receiving at a controller a request from a user interface, the controller in communication with the user interface;

identifying at the controller a process module associated with the request;

routing the request from the controller to the process module to invoke the process module;

in response to the input, invoking a the process module associated with a business process, the process module being invoked in one of a plurality of states, each state comprising logic defining a portion of the business process and comprising an identifier of a view to be presented to the user corresponding to the state, the view identifier comprising a logical specification of the corresponding view, the process module operable to provide the view identifier corresponding to the invoked state to the controller;

receiving at the controller the view identifier from the process module; and

generating at the controller the view corresponding to the invoked state at the user interface based on the logical specification in the view identifier.

9. (Previously Presented) The method of Claim 8, wherein generating the view comprises:

- determining a channel for the user interface;
- selecting a channel adapter corresponding to the channel; and
- providing the logical specification to the channel adapter, wherein the channel adapter generates the corresponding view based on the logical specification.

10. (Previously Presented) The method of Claim 8, wherein generating the view comprises:

- retrieving a map associating the logical specification of the corresponding view with a physical implementation of the view;
- determining the physical implementation of the view based on the map; and
- generating the physical implementation of the view at the user interface.

11. (Previously Presented) The method of Claim 8, wherein:
the user interface is a first user interface, the input is first input, the state is a first state, and the view identifier is a first view identifier; and

the method further comprises:

- receiving second input from a second user interface;
- in response to the second input, invoking the process module in a second state comprising a second view identifier;
- receiving the second view identifier from the process module at the controller;

and

- generating the corresponding view for the second state based on the logical specification in the second view identifier.

12. (Previously Presented) The method of Claim 8, wherein generating the view is further based on a user profile associated with the user interface.

13. (Previously Presented) The method of Claim 8, wherein invoking the process module comprises:

determining a characteristic of the user; and
invoking one of a plurality of process modules based on the characteristic of the user.

14. (Previously Presented) The method of Claim 13, wherein:

the characteristic is an experience level of the user; and
the characteristic is determined based on a user profile.

15. (Currently Amended) A system for generating a view at a user interface, comprising:

a user interface operable to receive input;

a controller in communication with a user interface and operable to to:

receive a request from the user interface;

identify a process module associated with the request; and

route the request to the process module to invoke the process module, invoke a process the process module associated with a business process in response to the input, the process module being invoked in one of a plurality of states, each state comprising logic defining a portion of the business process and comprising an identifier of a view to be presented to the user corresponding to the state, the view identifier comprising a logical specification of the corresponding view, the process module operable to provide the view identifier corresponding to the invoked state;

a view generator the controller operable to:

receive the view identifier from the process module; and

generate the view corresponding to the invoked state at the user interface based on the logical specification in the view identifier.

16. (Previously Presented) The system of Claim 15, wherein the view generator is further operable to:

- determine a channel for the user interface;
- select a channel adapter corresponding to the channel; and
- provide the logical specification to the channel adapter, wherein the channel adapter generates the corresponding view based on the logical specification.

17. (Previously Presented) The system of Claim 15, wherein:
the user interface is a first user interface, the input is first input, the state is a first state, and the view identifier is a first view identifier;

the system further comprises:

- a second user interface operable to receive second input; and
- a second view generator;

the controller is further operable to invoke the process module in a second state comprising a second view identifier in response to the second input; and

the second view generator is operable to receive the second view identifier from the process module at the controller and to generate the corresponding view for the second state based on the logical specification in the second view identifier.

18. (Previously Presented) The system of Claim 15, wherein the controller is further operable to:

- determine a characteristic of the user; and
- invoke one of a plurality of process modules based on the characteristic of the user.

19. (Previously Presented) The system of Claim 18, wherein:

- the characteristic is an experience level of the user; and
- the characteristic is determined based on a user profile.

20. (Previously Presented) The system of Claim 15, wherein the view generator is further operable to:

retrieve a map associating the logical specification of the corresponding view with a physical implementation of the view; and

determine the physical implementation of the view based on the map.